

CLAIMS

- 1 1. An apparatus comprising:
2 a plurality of processors, each processor having the capability of executing a
3 plurality of threads;
4 a memory coupled to the plurality of processors; and
5 a thread dispatch mechanism residing in the memory and executed by at least one
6 of the plurality of processors, the thread dispatch mechanism determining which of the
7 plurality of processors are idle, which of the plurality of processors can accept an
8 additional thread, and which of the plurality of processors cannot accept an additional
9 thread, the thread dispatch mechanism dispatching a new thread to an idle processor, if
10 one exists.
- 1 2. The apparatus of claim 1 wherein, if none of the plurality of processors is idle and
2 if at least one of the plurality of processors can accept an additional thread, the thread
3 dispatch mechanism dispatches the new thread to one of the plurality of processors that
4 can accept an additional thread.
- 1 3. The apparatus of claim 1 wherein, if all of the plurality of processors cannot
2 accept an additional thread, the thread dispatch mechanism waits for one of the plurality
3 of processors to complete processing a thread, thereby becoming a processor that can
4 accept an additional thread, and then dispatches the thread to the processor that can accept
5 an additional thread.

1 4. A method for dispatching threads in a computer system that includes a plurality of
2 processors that can each execute a plurality of threads, the method comprising the steps
3 of:

4 (1) determining the status of each of the plurality of processors, wherein a
5 processor is idle if not executing any threads, wherein the processor can accept an
6 additional thread if busy working on one or more threads but has the capacity to process
7 the additional thread, and wherein the processor cannot accept an additional thread if busy
8 working on a maximum number of threads the processor can execute; and

9 (2) dispatching a new thread to an idle processor, if one exists.

1 5. The method of claim 4 further comprising the step of:

2 if none of the plurality of processors is idle and if at least one of the plurality of
3 processors can accept an additional thread, the thread dispatch mechanism dispatches the
4 new thread to one of the plurality of processors that can accept an additional thread.

1 6. The method of claim 4 further comprising the steps of:

2 if all of the plurality of processors cannot accept an additional thread, the thread
3 dispatch mechanism waits for one of the plurality of processors to complete processing a
4 thread, thereby becoming a processor that can accept an additional thread, and then
5 dispatches the thread to the processor that can accept an additional thread.

1 7. A program product comprising:
2 (A) a thread dispatch mechanism that determines which of a plurality of
3 processors in a multiprocessor computer system are idle, which of the plurality of
4 processors can accept an additional thread, and which of the plurality of processors
5 cannot accept an additional thread, the thread dispatch mechanism dispatching a new
6 thread to an idle processor, if one exists, wherein each processor can execute a plurality
7 of threads; and
8 (B) computer-readable signal bearing media bearing the thread dispatch
9 mechanism.

1 8. The program product of claim 7 wherein the computer-readable signal bearing
2 media comprises recordable media.

1 9. The program product of claim 7 wherein the computer-readable signal bearing
2 media comprises transmission media.

1 10. The program product of claim 7 wherein, if none of the plurality of processors is
2 idle and if at least one of the plurality of processors can accept an additional thread, the
3 thread dispatch mechanism dispatches the new thread to one of the plurality of processors
4 that can accept an additional thread.

1 11. The program product of claim 7 wherein, if all of the plurality of processors
2 cannot accept an additional thread, the thread dispatch mechanism waits for one of the
3 plurality of processors to complete processing a thread, thereby becoming a processor that
4 can accept an additional thread, and then dispatches the thread to the processor that can
5 accept an additional thread.

* * * * *